AV1 5SBD

Create Table:

```
CREATE TABLE sbd-421821.bazar.temp_orders (
   order_id STRING,
   order_item_id STRING,
   purchase_date TIMESTAMP,
   payments_date TIMESTAMP,
   buyer_email STRING,
   buyer_name STRING,
   cpf STRING,
   buyer_phone_number STRING,
   sku STRING,
   product_name STRING,
   quantity_purchased INT64,
   currency STRING,
   item_price FLOAT64,
   ship_service_level STRING,
   recipient_name STRING,
   ship_address_1 STRING,
   ship_address_2 STRING,
   ship_address_3 STRING,
   ship_city STRING,
   ship_state STRING,
   ship_postal_code STRING,
   ship_country STRING,
   ioss_number STRING
);

CREATE TABLE sbd-421821.bazar.orders (
   order_id STRING,
   purchase_date TIMESTAMP,
   payments_date TIMESTAMP,
   buyer_email STRING,
   ship_service_level STRING,
   recipient_name STRING,
   ship_address_1 STRING,
   ship_address_2 STRING,
   ship_address_3 STRING,
   ship_city STRING,
   ship_state STRING,
   ship_postal_code STRING,
   ship_country STRING
);

CREATE TABLE sbd-421821.bazar.order_items (
   order_id STRING,
   order_item_id STRING,
   sku STRING,
   product_name STRING,
   quantity_purchased INT64,
   currency STRING,
   item_price FLOAT64
);

CREATE TABLE sbd-421821.bazar.customers (
   buyer_email STRING,
   buyer_name STRING,
   cpf STRING,
   buyer_phone_number STRING
);

CREATE TABLE sbd-421821.bazar.products (
   sku STRING,
   product_name STRING
);

CREATE TABLE sbd-421821.bazar.stock_movement (
   sku STRING,
   order_id STRING,
   quantity_purchased INT64,
   movement_date TIMESTAMP
);

CREATE TABLE sbd-421821.bazar.purchases (
   sku STRING,
   quantity_to_purchase INT64,
   purchase_date TIMESTAMP
);
```

Procedure para carregar os pedidos na tabela de carga temporária:

```
CREATE OR REPLACE PROCEDURE load_temp_orders(csv_data STRING)
BEGIN
  -- Carregar os dados CSV para a tabela temporária
  CREATE TEMP TABLE temp_orders AS
  SELECT *
  FROM CSV.EMBED(
    @csv_data,
    'order_id STRING, order_item_id STRING, purchase_date TIMESTAMP,
payments_date TIMESTAMP,
     buyer_email STRING, buyer_name STRING, cpf STRING, buyer_phone_number
STRING,
     sku STRING, product_name STRING, quantity_purchased INT64, currency
STRING,
     item_price FLOAT64, ship_service_level STRING, recipient_name STRING,
     ship_address_1 STRING, ship_address_2 STRING, ship_address_3 STRING,
     ship_city STRING, ship_state STRING, ship_postal_code STRING, ship_country
STRING, ioss_number STRING'
  );
END;
```

Procedure para carregar os dados das tabelas de Pedidos, Itens Pedido:

Clientes e Produtos a partir da tabela temporária:

```
CREATE OR REPLACE PROCEDURE process_orders()
BEGIN
  -- Inserir clientes novos na tabela de clientes
  INSERT INTO bazar.customers (buyer_email, buyer_name, cpf,
buyer_phone_number)
  SELECT DISTINCT buyer_email, buyer_name, cpf, buyer_phone_number
  FROM temp_orders
  WHERE buyer_email NOT IN (SELECT buyer_email FROM bazar.customers);
  -- Inserir produtos novos na tabela de produtos
  INSERT INTO bazar.products (sku, product_name)
  SELECT DISTINCT sku, product_name
  FROM temp_orders
  WHERE sku NOT IN (SELECT sku FROM bazar.products);

  -- Inserir pedidos na tabela de pedidos
  INSERT INTO bazar.orders (order_id, purchase_date, payments_date,
buyer_email, ship_service_level,
                         recipient_name, ship_address_1, ship_address_2,
ship_address_3,
                         ship_city, ship_state, ship_postal_code, ship_country)
  SELECT DISTINCT order_id, purchase_date, payments_date, buyer_email,
ship_service_level,
                recipient_name, ship_address_1, ship_address_2, ship_address_3,
                ship_city, ship_state, ship_postal_code, ship_country
  FROM temp_orders;

  -- Inserir itens de pedido na tabela de itens de pedido
  INSERT INTO bazar.order_items (order_id, order_item_id, sku, product_name,
quantity_purchased, currency, item_price)
  SELECT order_id, order_item_id, sku, product_name, quantity_purchased,
currency, item_price
  FROM temp_orders;
END;
```

Procedure para atualizar a tabela de movimentação de estoque:

```
CREATE OR REPLACE PROCEDURE update_stock_movement()
BEGIN
  -- Atualizar a tabela de movimentação de estoque
  INSERT INTO bazar.stock_movement (sku, order_id, quantity_purchased,
movement_date)
  SELECT t.sku, t.order_id, t.quantity_purchased, CURRENT_TIMESTAMP()
  FROM temp_orders t
  LEFT JOIN bazar.order_items oi ON t.order_id = oi.order_id AND t.sku = oi.sku
  ORDER BY t.item_price DESC; -- Ordenar do maior valor para o menor
```

```
END;
```

Procedure para atualizar o estoque dos produtos:

```
CREATE OR REPLACE PROCEDURE update_product_stock()
BEGIN
  -- Atualizar estoque dos produtos
  UPDATE bazar.products p
  SET p.stock = p.stock + s.quantity_purchased
  FROM bazar.stock_movement s
  WHERE p.sku = s.sku;
END;
```